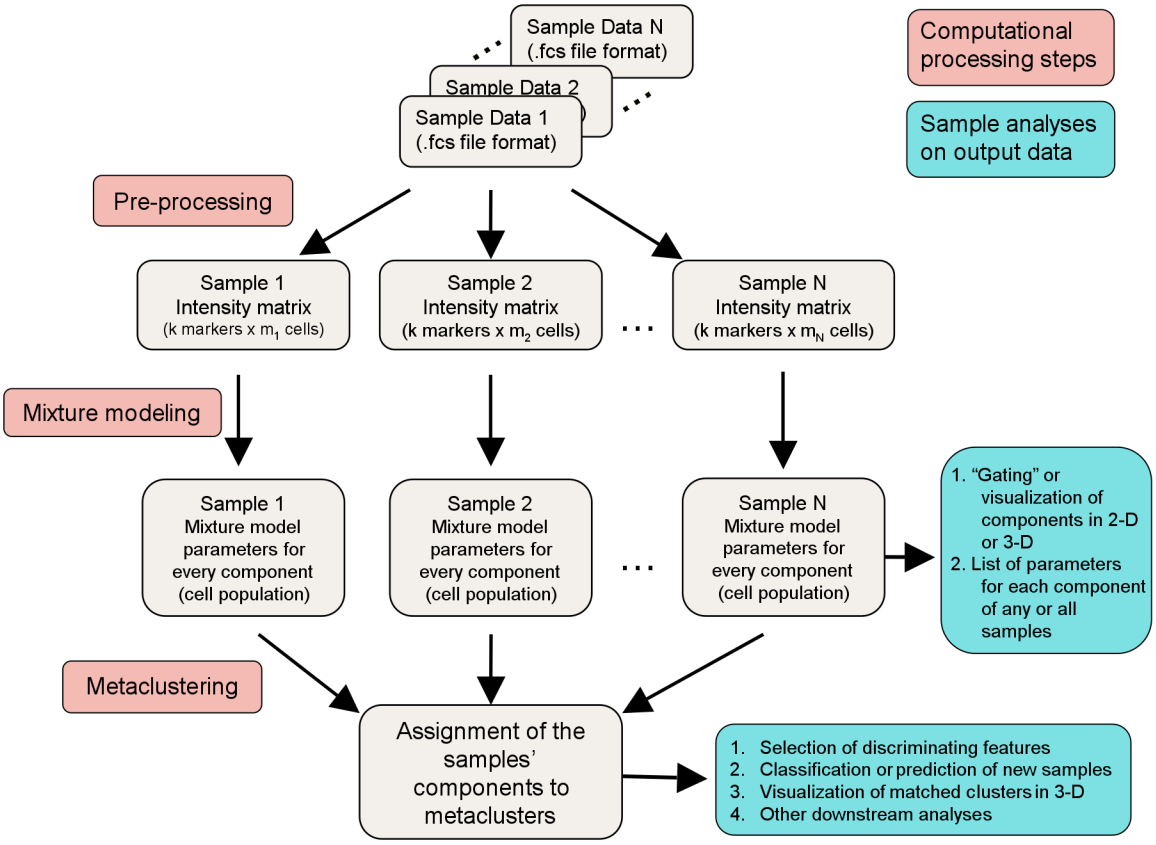
Supplementary Information: Automated High-dimensional Flow Cytometric Data Analysis

Saumyadipta Pyne[1], Xinli Hu[1♣], Kui Wang[3♣], Elizabeth Rossin[1♣], Tsung-I Lin[4], Lisa M. Maier[1,2], Clare Baecher-Allan[2], Geoffrey J. McLachlan[5], Pablo Tamayo[1], *David A. Hafler[1,2], Philip L. De Jager[1,2,6†], and *Jill P. Mesirov[1†].

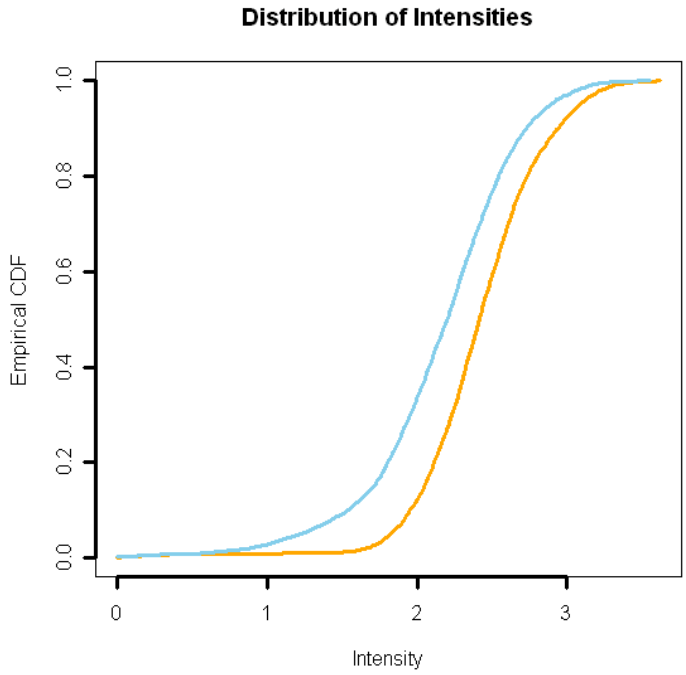
[1] Broad Institute of Massachusetts Institute of Technology and Harvard University, 7 Cambridge Center, Cambridge MA 02142, USA.
[2] Division of Molecular Immunology, Center for Neurologic Diseases, Brigham & Women's Hospital and Harvard Medical School, 77 Avenue Louis Pasteur, Boston, MA 02115, USA.
[3] Department of Mathematics, University of Queensland, St. Lucia, Queensland, 4072, Australia.
[4] Department of Applied Mathematics, National Chung Hsing University, Taichung 402, Taiwan.
[5] Department of Mathematics & Institute for Molecular Bioscience, University of Queensland, St. Lucia, Queensland, 4072, Australia.
[6] Partners Center for Personalized Genetic Medicine, Boston, MA 02115, USA.

♣ These authors contributed equally to this work
† These authors contributed equally to this work
* Corresponding authors: David Hafler, hafler@broad.mit.edu, and Jill Mesirov, mesirov@broad.mit.edu

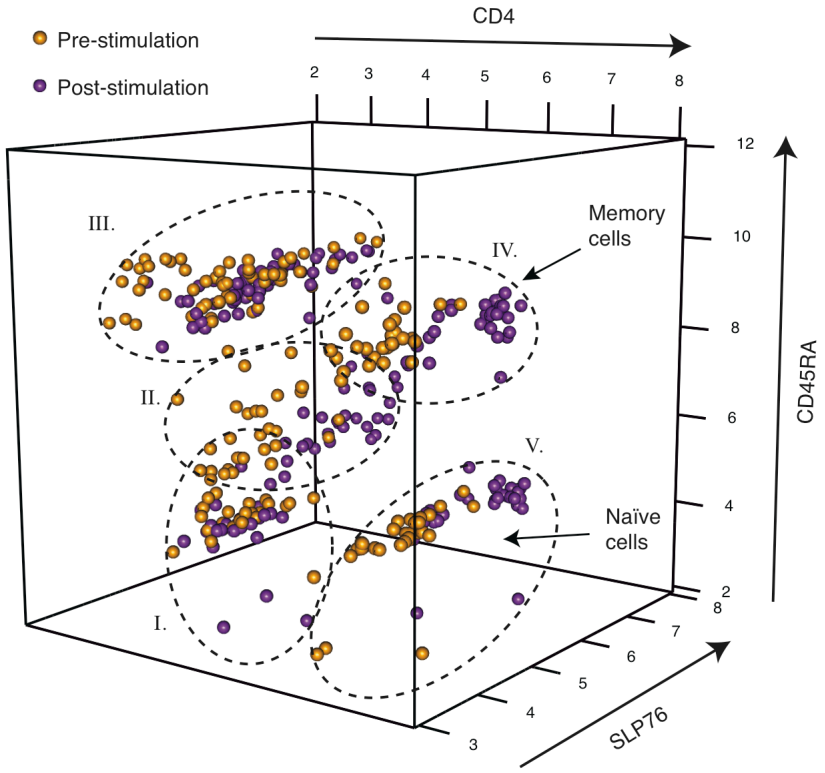
Supplementary Figures and Tables



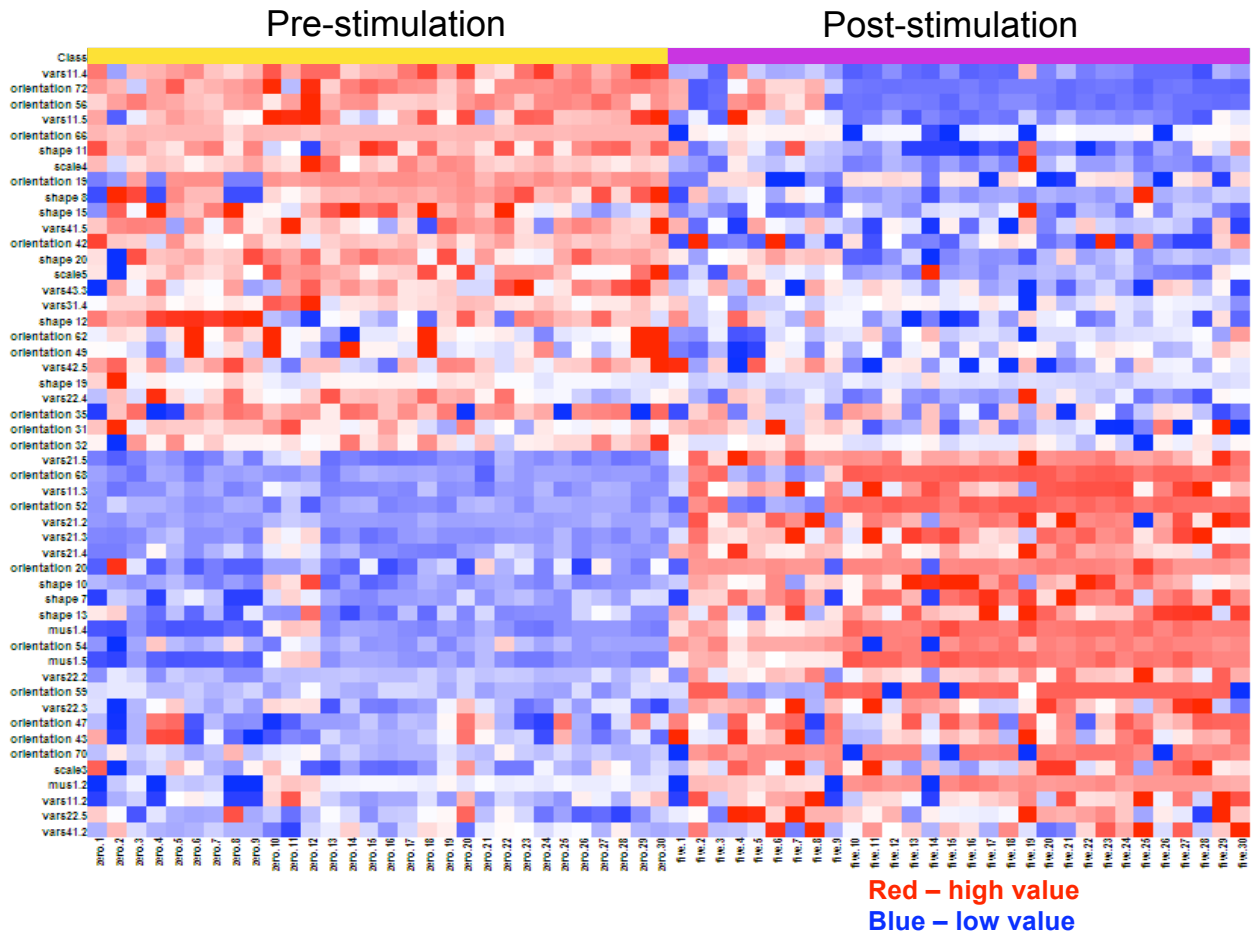
Supplementary Figure S1. Schematic representation of FLAME dataflow. In this flowchart, we outline the dataflow for FLAME's computational pipeline beginning with the raw flow cytometric data files (in .fcs format) for all the samples and ending with the assignment of their components to metaclusters. At each stage of the process we indicate what external analyses or visualizations can be done with the intermediate data (output files). FLAME processing steps are noted in salmon, external functions are noted in turquoise.



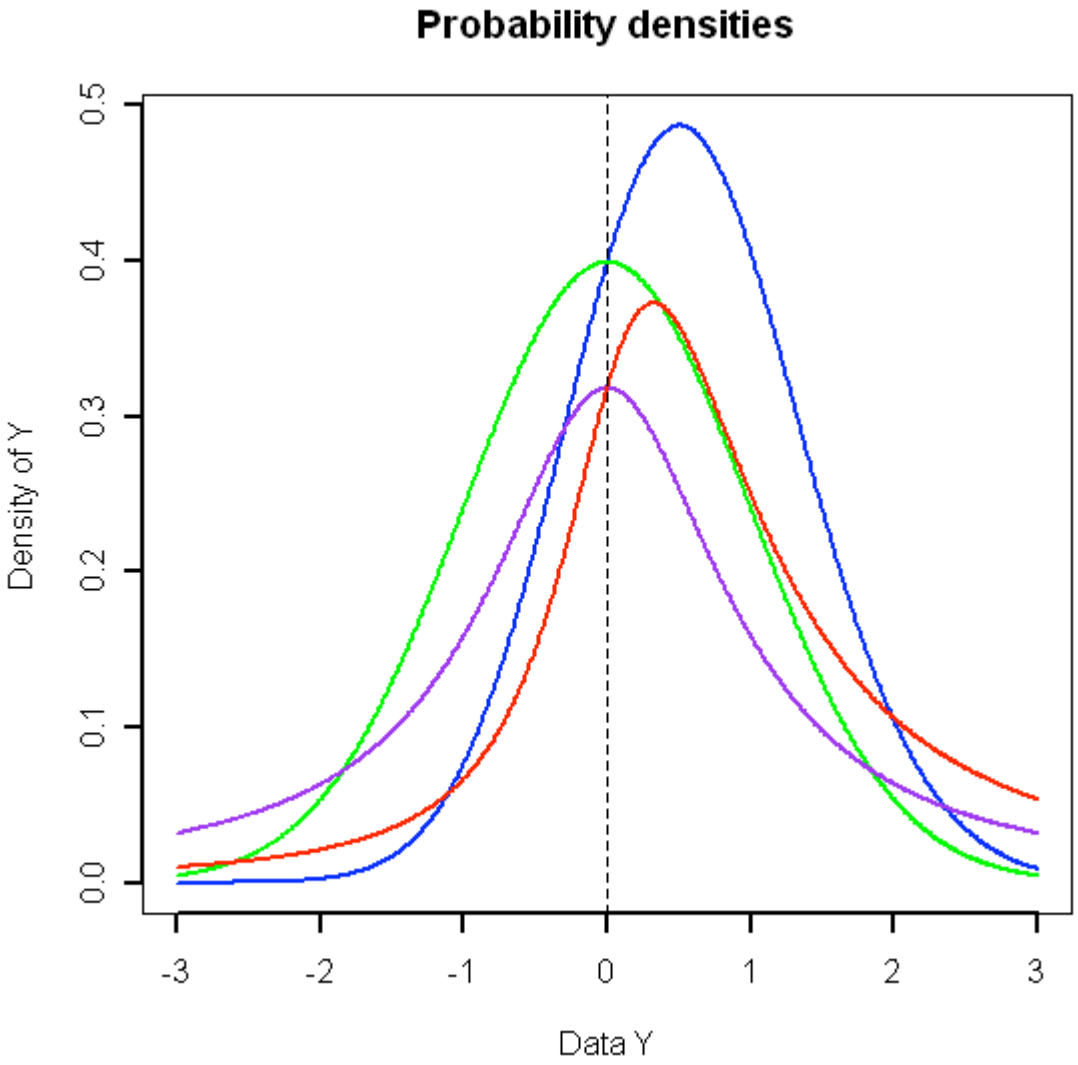
Supplementary Figure S2. Expression values for the HLA-DQ and CD95 markers in a representative lymphoblastic cell line indicate unimodal distributions of expression for either marker. In plot (a), the empirical cumulative distributions of the two dimensions are plotted: HLA DQ in blue and CD95 in orange. The smooth ascent of both distributions from 0 to 1 is indicative of unimodal density of expression for either marker.



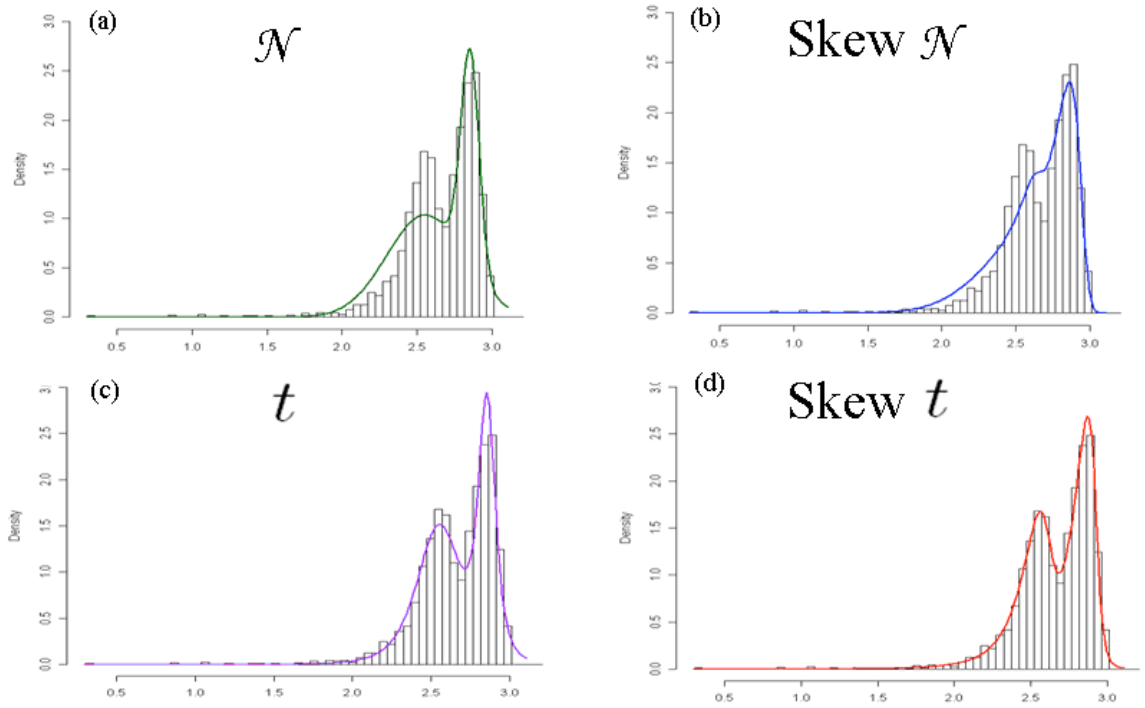
Supplementary Figure S3. Meta-clustering of cell populations across 29 subjects measured before and after T cell receptor stimulation. Results of FLAME's use of PAM to match subjects' clusters across the cohort. First, the pre- (yellow) and post- (purple) stimulation modes for each cohort were metaclustered independently – each cohort yielding five populations. Next, the corresponding metaclusters between the two classes were identified. In the figure, all modes from all subjects are overlaid to illustrate the five subpopulations and the difference in phosphorylation after stimulation.



Supplementary Figure S4. Classifying pre- and post-stimulation samples by distinctive mixture model features. The heat map, based on the results of a feature selection exercise using paired *t* test across pre- and post-stimulation classes, shows a set of features that are most distinctive across the pre- and post-stimulation samples. Each column of the heat map represents one of the 58 samples interrogated in this experiment, and each row presents data from one of the top 50 discriminating features. The pre-stimulation samples (0 minutes, yellow subset) are depicted on the left half of the heat map and the post-stimulation samples (5 minutes, purple subset) on the right. Features with high pre-stimulation (red) and low post-stimulation values (blue) in general are observed in the upper half of the heat map. The lower half of the heat map contains those features with the opposite pattern of changes in parameters. For details about the selected features, see Supplementary Table T1.



Supplementary Figure S5. Different choices of probability density functions in FLAME. The forms of t, skew normal and skew t densities are plotted in violet, blue and red respectively. The standard normal (or Gaussian) density plot in green is also included for reference. In this example, all densities have location parameter 0; skew t and skew normal have skew shape parameter equal to 1; skew t and t have one degree of freedom. Although FLAME uses multivariate distributions, we show univariate forms in this plot for convenient visualization.



Supplementary Figure S6. Mixture modeling with different distributions. Here we fit a skewed one-dimensional intensity distribution (unpublished data, from M.G. Kharas and D.G. Gilliland) with (a) normal, (b) skew normal, (c) t and (d) skew t mixture models plotted in green, blue, violet and red respectively. While all four distributions yield 2-component univariate models, skew t provides the best fit to the actual distribution.

Supplementary Table T1: List of the 50 most significantly distinctive features when comparing the pre- and post-stimulation samples of lymphocytes with anti-CD3 in Figure 3. A "feature" in this case is a parameter describing one property of a cell cluster found in the samples being investigated. For example, the most differentiated parameter after anti-CD3 stimulation is "mus1.4" (P=6.12x10^{-22}); this is the mode of the intensity in dimension 1 for cluster 4 (which corresponds to naïve T cells). For every feature, its ID, type, cluster number, dimension(s), change after 5 minutes of stimulation (mean(5)-mean(0)) and the corresponding P-value (from paired t test across pre- and post-stimulation classes) are reported. (A list for the terminology of parameters and statistics computed by FLAME is available with the software.)

Feature ID	Feature Type	Cluster #	Dimension(s)	mean(5)-mean(0)	p-value
mus1.4	Mean	4	1	1.761	6.13E-22
mus1.5	Mean	5	1	1.657	2.47E-21
vars21.5	Variance	5	1,2	0.088	6.67E-21
mus1.2	Mean	2	1	1.571	1.52E-18
vars11.4	Variance	4	1	-0.156	1.65E-18
shape 7	Shape	2	2	0.682	4.49E-16
orientation 72	Orientation	5	3	-0.649	1.01E-14
orientation 56	Orientation	4	3	-0.609	1.13E-12
orientation 68	Orientation	5	1	0.538	1.83E-12
vars11.3	Variance	3	1	0.314	6.22E-12
vars21.3	Variance	3	1,2	0.259	1.14E-11
orientation 20	Orientation	2	1	0.504	2.17E-11
vars21.2	Variance	2	1,2	0.251	1.22E-10
orientation 52	Orientation	4	1	0.552	1.87E-10
shape 10	Shape	3	3	0.740	1.31E-09
shape 13	Shape	4	4	1.023	4.37E-09
shape 8	Shape	2	2	-0.141	4.41E-09
orientation 54	Orientation	4	2	0.534	1.26E-08
shape 11	Shape	3	3	-0.175	2.62E-08
vars21.4	Variance	4	1,2	0.060	3.42E-08
vars11.5	Variance	5	1	-0.082	4.00E-08
shape 15	Shape	4	4	-0.178	5.17E-07
orientation 19	Orientation	2	1	-0.632	1.34E-06
scale4	Scale	4	NA	-0.052	3.32E-06
vars22.3	Variance	3	2	0.146	1.09E-05
orientation 66	Orientation	5	1	-0.515	1.37E-05
vars41.5	Variance	5	1,4	-0.024	2.63E-05
orientation 47	Orientation	3	4	0.561	4.65E-05
vars22.2	Variance	2	2	0.282	5.45E-05
shape 20	Shape	5	4	-0.060	7.93E-05
orientation 43	Orientation	3	3	0.066	8.01E-05
orientation 59	Orientation	4	3	0.548	1.51E-04
vars11.2	Variance	2	1	0.131	1.62E-04
scale3	Scale	3	NA	0.063	2.19E-04

vars22.5	Variance	5	2	0.023	2.65E-04
shape 12	Shape	3	4	-0.073	4.58E-04
vars43.3	Variance	3	3,4	-0.020	7.10E-04
scale5	Scale	5	NA	-0.038	7.23E-04
orientation 42	Orientation	3	3	-0.422	9.73E-04
vars42.5	Variance	5	2,4	-0.014	3.32E-03
vars31.4	Variance	4	1,3	-0.015	3.34E-03
vars41.2	Variance	2	1,4	0.099	3.42E-03
orientation 70	Orientation	5	2	0.308	4.07E-03
orientation 62	Orientation	4	4	-0.264	4.55E-03
orientation 49	Orientation	4	1	-0.234	4.87E-03
vars22.4	Variance	4	2	-0.028	1.98E-02
orientation 35	Orientation	3	1	-0.267	2.23E-02
orientation 31	Orientation	2	3	-0.040	4.46E-02
shape 19	Shape	2	1	-0.231	4.89E-02
orientation 32	Orientation	2	4	-0.030	5.06E-02

Supplementary Methods

Details of the datasets presented in the manuscript

The three datasets were generated as part of other efforts. Here, we present the pertinent references and summarize the key details to provide the biological context of each experiment as well as the manner in which data were generated and handled prior to upload into FLAME. In each case, compensation was performed at the time of data collection on each flow cytometer.

1. *Lymphoblastic cell line data*. 194 LCLs – each generated from a different individual - were cultured in three batches and stained with anti-HLA DQ and anti-CD95 antibodies as described in detail in a recent manuscript (D. Altshuler, personal communication). In brief, data on up to 5000 cells (minimum 500 cells) were captured by a BD Biosciences FACSCalibur system, and a .fcs file was generated for each cell line. This file was first pre-processed with FLAME using the forward scatter (FSC, cell size) and side scatter (SSC, cell granularity) dimensions of information to resolve the population of live cells from dead cells and cellular debris that are found in all cell cultures. Data on the population of live cells were then saved into a new .fcs file and processed in the standard manner described in the methods section of the main text: the data underwent a logicle[1] transformation prior to being uploaded into the FLAME software for modeling.

2. *Regulatory T cell data*. In this dataset, a sample of peripheral blood was processed using Ficoll extraction to segregate PBMCs[2]. PBMCs were then stained with fluorophore-labeled antibodies against CD4, CD25, HLA DR, and Foxp3 as described elsewhere[2]. Data were then captured using a BD Biosciences FACSAria system. Flowjo[3] was then used to project the data in the FSC and SSC dimensions, and a human operator gated the live PBMC cells and saved the reduced dataset into a *.fcs file. A Logicle transformation was then applied to these data before uploading into FLAME.

3. *T cell phosphorylation data*. These data have been previously published, and a detailed description of the generation and processing of these data is presented elsewhere [4]. In brief, Maier and colleagues captured data on whole blood stained with fluorophore-labeled antibodies against CD4, CD45RA, SLP76 (pY128) and ZAP70 (pY292) before and after stimulation with an anti-CD3 antibody. For each subject, one blood sample was stained prior to anti-CD3 stimulation in whole blood. A second sample was stained 5 minutes after stimulation. Data were captured using a BD Biosciences FACSCalibur system. The .fcs files were then pre-processed using Flowjo[3], and the operator gated the lymphocyte population of each sample. This reduced dataset was then used to generate parameters for the different cell populations under study. To enable a comparison of FLAME with these manual results, we generated a .fcs files that contained the cells found within the lymphocyte gate defined by the manual operator. Data in

these files then underwent logicle transformation before being uploaded into the FLAME software.

Details of the FLAME mixture modeling

Finite mixture models have been extensively developed and widely applied to clustering, classification, density estimation, and pattern recognition problems, as shown by McLachlan and Basford[5], McLachlan and Peel[6], and Frühwirth-Schnatter[7], and the references therein. Although Gaussian mixture modeling has enjoyed widespread use in numerous past applications, the tails of the Gaussian distribution are often found to be shorter than required in the presence of outlier events. In recent years the use of finite mixture of *t* densities has steadily gained acceptance for providing robust modeling based on the properties of the *t* distribution[6, 8]. Also of recent origin, the multivariate *skew normal* and *skew t* distributions have been shown to be beneficial in dealing with asymmetric data in various theoretical and applied problems[9-11]. The distributional properties and stochastic representations of multivariate *skew normal* and *t* models are detailed in Gupta[12] and Gupta et al.[13]. Some extensions of *skew normal* and *t* models are discussed in Azzalini et al.[14] and Sahu et al[15].

In many applied problems the contours of the clusters may be distorted, and inferences based on symmetric normal or *t* mixture models can be misleading when the data involve highly asymmetrically distributed observations. In particular, the normal/*t* mixture model or its generalized version[16] tends to split and produce many clusters spuriously, as additional components are needed to accommodate the skew and asymmetry in the data. An increase in the number of pseudo-components can lead to difficulty in interpretation of results as well as cause inefficient computation. To address these situations, mixtures of multivariate *skew normal*/*t* distributions are required; however the use of EM algorithms for multivariate *skew normal*/*t* mixture models has been very limited in the literature because of the complexity of their implementation.

Only recently, Lin et al.[17] and Lin et al.[18] have proposed univariate *skew normal* and *skew t* mixture models. Our paper is the first to use mixtures of multivariate *skew t* components. The complete-data framework for the EM algorithm for this problem does not automatically translate into a manageable multivariate version. In particular, the conditional expectations on the E-step can no longer be carried out in closed form, and then the equations on the M-step cannot be solved by extending the methods used to solve iteratively the univariate equations. We circumvented these problems on the E- and M-steps by proposing a new EM framework in which to implement the EM algorithm by adopting a different characterization of the multivariate t-distribution, namely a variant of the one proposed in Sahu et al[15] (*Canadian J Stat*., 31, 129-150 (2003)). This allowed us to effect the E-step in closed form and to provide a set of equations on the M-step that can almost be solved in closed form apart for the one for the component-degrees of freedom. Below we give the details of our multivariate *skew t* mixture modeling algorithm, preceded by the alternate modeling options in FLAME, the multivariate *t* and *skew normal* mixture models.

Multivariate t Mixture Model

The multivariate *t* distribution: As explained in McLachlan and Peel[6], the multivariate *t* distribution can be characterized as follows. For a fixed scalar λ, suppose that $Y \sim N(\xi, \Sigma/\lambda)$, which is a multivariate normal distribution with mean vector ξ, and covariance matrix Σ/λ. If we assume λ follows a gamma distribution, $\lambda \sim \text{gamma}(\nu/2, \nu/2)$, then the unconditional distribution of *Y* defines the multivariate *t* distribution with location parameter ξ, positive definite scale matrix Σ, and degrees of freedom ν (ν>2), which is given by

$$f(y;\xi,\Sigma,\nu) = \frac{\Gamma(\frac{\nu+k}{2})\,|\Sigma|^{-1/2}}{(\pi\nu)^{k/2}\Gamma(\frac{\nu}{2})\{1+\eta/\nu\}^{(k+\nu)/2}} \qquad (1)$$

where $\eta = (y-\xi)^T\Sigma^{-1}(y-\xi)$ denotes the Mahalanobis squared distance between y and ξ with Σ as the scale matrix. For the multivariate *t* distribution, it can be shown that the mean and covariance matrix are given by $E(Y) = \xi$, $\text{cov}(Y) = \frac{\nu}{\nu-2}\Sigma$.

The multivariate *t* mixture model: With a mixture model-based approach for unsupervised learning, the *k*-dimensional observed-data points (often called *feature vectors*) $Y_1,\ldots,Y_n$ are assumed to have come from a mixture of *g* components in some unknown non-negative mixing proportions $p_1,\ldots p_g$ which sum to one. The number of components in this mixture model corresponds to the number of clusters to be imposed on the data. A common practice is to use component distributions belonging to the same parametric family, in this case, the multivariate *t* distribution (1). In this case, each feature vector is taken to be a realization of the mixture probability density function

$$f(y,\Psi) = \sum_{h=1}^{g} p_h f_h(y;\xi_h,\Sigma_h,\nu_h)$$

where $f_h(y;\xi_h,\Sigma_h,\nu_h)$ denotes the h^{th} *k*-dimensional *t* component with location parameter ξ_h, scale matrix Σ_h, and degrees of freedom ν_h. The vector of unknown parameters is denoted by Ψ and can be estimated by the maximum likelihood (ML) method via the EM[19].

The EM algorithm: For application of the EM algorithm, the observed-data vector $(y_1^T,\ldots,y_n^T)^T$ is regarded as incomplete. Using the representation of the multivariate *t* distribution (1), we include λ_h as an unknown "latent" variable where, conditional on the y_j's and membership of the h^{th} component, the distribution of Y_j can be taken to be multivariate normal with mean $\xi$and covariance matrix Σ_h/λ_h (*h*=1…*g*).

The component-label indicator variables z_{jh} are introduced, where z_{jh} is defined to be one or zero based on whether y_j did or did not arise from the h^{th} component of the mixture model (*h*=1,...,*g*; *j*=1,...,*n*). Letting $z_j = (z_{j1},\ldots,z_{jg})^T$, the complete-data vector

x_c is given by $x_c = (x_1^T, ..., x_n^T)$, where $x_1 = (y_1^T, \lambda_1^T, z_1^T)^T, ..., x_n = (y_n^T, \lambda_n^T, z_n^T)^T$ are taken to be independent and identically distributed with $z_1, ..., z_n$ being independent realizations from a multinomial distribution consisting of one draw on g categories with respective probabilities $p_1, ..., p_g$. That is,

$p_1, ..., p_g \sim Mult_g(1, p)$, where $p = (p_1, ..., p_g)^T$.

For this specification, the complete-data log likelihood is

$$l_c = \sum_{j=1}^{n} \sum_{h=1}^{g} \{-0.5[k \log(2\pi) + \log(|\Sigma_h| / \lambda_h) + \lambda_h (y_j - \xi_h)^T \Sigma_h^{-1} (y_j - \xi_h)]$$
$$+ [-\lambda_h \nu_h / 2 + \nu_h \log(\nu_h / 2) / 2 - \log(\Gamma(\nu_h / 2) + (\nu_h / 2 - 1) \log(\lambda_h)] + \log(p_h)\} z_{jh}$$

The EM algorithm proceeds iteratively in two steps: E step and M step.

The E step comprises of computing the following conditional expectations, using the current fit for the vector of unknown parameters Ψ:

$$E(z_{jh} \mid y_j) = \tau_{jh} = \frac{p_h f_h(y_j; \xi_h, \Sigma_h, \nu_h)}{\sum_h p_h f_h(y_j; \xi_h, \Sigma_h, \nu_h)},$$
$$E(\lambda_h \mid y_j, z_{jh} = 1) = e_{jh} = \frac{\nu_h + k}{\nu_h + (y_j - \xi_h)^T \Sigma_h^{-1} (y_j - \xi_h)},$$

while the M step updates the estimates of the parameters, using the equations

$$p_h = \frac{1}{n} \sum_{j=1}^{n} \tau_{jh},$$
$$\Sigma_h = \sum_{j=1}^{n} (y_j - \xi_h)(y_j - \xi_h)^T e_{jh} \tau_{jh} / \sum_{j=1}^{n} \tau_{jh},$$
$$\xi_h = \sum_{j=1}^{n} y_j e_{jh} \tau_{jh} / \sum_{j=1}^{n} (e_{jh} \tau_{jh}).$$

The E and M steps are alternated repeatedly until the likelihood changes by a predefined arbitrarily small amount and the process has reached convergence.

Multivariate Skew Normal Mixture Model

The multivariate *skew normal* distribution: As developed by Azzalini[9, 11], a random variable *Y* follows a univariate *skew normal* (*SN*) distribution with location parameter ξ, scale parameter σ^2, and skewness parameter λ if it has the density

$$2\phi(\frac{y - \xi}{\sigma})\Phi(\lambda \frac{y - \xi}{\sigma}) / \sigma$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the density function and cumulative distribution function, respectively, for the standard normal distribution; then, for brevity, we say that $Y \sim SN(\xi, \sigma^2, \lambda)$. Note that if $\lambda = 0$, the density of *Y* reduces to the $N(\xi, \sigma^2)$ density.

A *k*-dimensional extension of (1) is given in Azzalini and Dalla Valle[11]. It was constructed via a transformation method as follows: $U=(U_1,...,U_k)^T \sim N_k(0, \Xi)$ with standardized marginals, independent of $U_0 \sim N(0,1)$; if $(\delta_1 \ldots \delta_k)$ are in (-1, 1), define $Y_j = \delta_j |U_0| + (1-\delta_j^2)U_j$ $(j=1,...,k)$. Then the vector $Y=(Y_1,...,Y_k)^T$ has a *k*-dimensional *skew normal* distribution with density function

$$2\phi_k(y;\xi=0,\Omega)\Phi(\alpha y), \qquad y \in R^k \tag{2}$$

where $\phi_k(y;\xi,\Omega)$ is the density function of the *k*-dimension normal distribution with mean ξ, and covariance Ω; α and Ω are functions of $(\delta_1,\ldots,\delta_k)$ and Ξ [11]. An alternative definition was discussed in Gupta and Chen[13].

Now we come to our definition of the multivariate *skew normal* distribution. Note that Ξ is the correlation matrix of *U*, when standardized. In practice, the means and variances/covariances are typically the main interest of the analysis. To extend (2) to a general situation where *U* is defined with its location parameter (mean) ξ, and scale matrix Σ, we proceed similarly as in Sahu et al.[15]. Let $\boldsymbol{\delta}$ be a *k*-dimensional vector, $U=(U_1,...,U_k)^T \sim N_k(\xi, \Omega)$, $U_0 \sim N(0,1)$, then $Y = \delta |U_0| + U$ defines a variant of the *skew normal* distribution (2) with its density function equal to

$$2\phi_k(y;\xi,\Omega)\Phi(\alpha(y-\xi)), \qquad y \in R^k, \tag{3}$$

where $\Omega = \Sigma + \delta\delta^T$, $\alpha = \delta^T\Omega^{-1}/(1-\delta^T\Omega^{-1}\delta)^{1/2}$.

Note that in the definition of *Y* in Sahu et al.[15], the random coefficient of each element of $\boldsymbol{\delta}$ is allowed to be different.

For the multivariate *skew normal* distribution (3), we provide the first two moments. These are obtained from the moment generating function

$$M(s) = E_Y(e^{y^T s}) = 2\Phi(\delta^T s)\exp(\frac{1}{2}s^T\Omega s + \xi^T s) .$$

The derivation of this equation is similar to that of Sahu et al.[15]. The mean and covariance matrix are given by $E(Y) = \xi + \sqrt{\frac{2}{\pi}}\delta$ and $\mathrm{cov}(Y) = \Sigma + (1-\frac{2}{\pi})\delta\delta^T$.

The multivariate *skew normal* mixture model: Like earlier, we assume the component distributions belong to the same parametric family, in this case, the *skew normal* distribution (3). Using similar notation as above, the mixture probability density function is denoted by

$$f(Y,\Psi) = \sum_{h=1}^{g} p_h f_h(y;\xi_h,\Sigma_h,\delta_h),$$

where $f_h(y;\xi_h,\Sigma_h,\delta_h)$ denotes the h^{th} *k*-dimensional *skew normal* component with location parameter ξ_h, scale matrix Σ_h and skew parameter δ_h. The vector of unknown

parameters is denoted by Ψ and can be estimated by the maximum likelihood method via the EM algorithm [19]. Recently the model has also received attention from other groups[20].

The EM algorithm: Again we re-use much of the earlier notation here. For application of the EM algorithm, the observed-data vector $(y_1^T,...,y_n^T)^T$ is regarded as being incomplete. Using the representation of *skew normal* distribution (3), we include *v* as a latent unobservable variable. The component-label indicator variables z_{jh} are also introduced, where z_{jh} is defined to be one or zero according to if y_j did or did not arise from the *h* th component of the mixture model, (*h*=1...*g*; *j*=1...*n*). Letting $z_j = (z_{j1},...,z_{jg})^T$, the complete-data vector x_c is given by $x_c = (x_1^T,...,x_n^T)$ where $x_1 = (y_1^T, u_1^T, z_1^T)^T,...,x_n = (y_n^T, u_n^T, z_n^T)^T$ are taken to be independent and identically distributed with $z_1,...,z_n$ being independent realizations from a multinomial distribution consisting of one draw on *g* categories with respective probabilities $p_1,...,p_g$. That is,

$p_1,...,p_g \sim Mult_g(1,p)$, where $p = (p_1,...,p_g)^T$.

For this specification, the complete-data log likelihood is

$$l_c = const - \sum_{i=1}^{n}\sum_{h=1}^{g}\{0.5[\log(|\Sigma_h|) + (y_i - \xi_h - \delta_h u_i)^T \Sigma_h^{-1}(y_i - \xi_h - \delta_h u_i) + u_i^2] + \log(p_h)\}z_{ih}$$

The EM algorithm proceeds iteratively in two steps: E-step and M-step.

The E step comprises of computing the following conditional expectations, using the current fit for the vector of unknown parameters:

$$E(z_{jh} \mid y_j) = \tau_{jh} = \frac{p_h f_h(y_j;\xi_h,\Sigma_h,\delta_h)}{\sum_h p_h f_h(y_j;\xi_h,\Sigma_h,\delta_h)},$$

$$\mu_{jh} = \delta_h^T(\Sigma_h + \delta_h\delta_h^T)^{-1}(y_j - \xi_h),$$

$$\sigma_h^2 = 1 - \delta_h^T(\Sigma_h + \delta_h\delta_h^T)^{-1}\delta_h,$$

$$E(u_j \mid y_j, z_{jh} = 1) = e_{1jh} = \mu_{jh} + \sigma_h\phi(\mu_{jh}/\sigma_h)/\Phi(\mu_{jh}/\sigma_h),$$

$$E(u_j^2 \mid y_j, z_{jh} = 1) = e_{2jh} = \mu_{jh}^2 + \sigma_h^2 + \mu_{jh}\sigma_h\phi(\mu_{jh}\sigma_h)/\Phi(\mu_{jh}\sigma_h),$$

while the M step updates the estimates of the parameters, using the following equations:

$$p_h = \frac{1}{n}\sum_{j=1}^{n}\tau_{jh},\ \delta_h = \frac{\sum_{j=1}^{n}\tau_{jh}e_{1jh}(y_j - \xi_h)}{\sum_{j=1}^{n}\tau_{jh}e_{2jh}},$$

$$\Sigma_h = \sum_{j=1}^{n}[(y_j - \xi_h)(y_j - \xi_h)^T - e_{1jh}\delta_h(y_j - \xi_h)^T - (y_j - \xi_h)\delta_h e_{1jh}^T) + e_{2jh}\delta_h\delta_h^T)]$$

$$\tau_{jh}/\sum_{j=1}^{n}\tau_{jh},$$

$\xi_h = \sum_{j=1}^{n}(y_j - \delta_h e_{1jh})\tau_{jh} / \sum_{j=1}^{n}\tau_{jh}$.

The E and M steps alternate repeatedly until the likelihood changes by a predefined arbitrary small amount, at which stage the process is deemed to have reached convergence.

Multivariate Skew t Mixture Model

The multivariate *skew t* distribution: We proceed similarly as in the case of the *skew normal* distribution, and again adopt the approach of Sahu et al.[15]. We let $\boldsymbol{\delta}$ be a *k*-dimensional vector, and suppose that conditional on *w*,

$$\begin{pmatrix} U_0 \\ U \end{pmatrix} \sim N\left(\begin{pmatrix} \xi \\ 0 \end{pmatrix} \quad \begin{pmatrix} \Sigma & 0 \\ 0 & 1 \end{pmatrix}\frac{1}{w}\right),$$

where $w \sim \text{gamma}\ (v/2, v/2)$.

Then $Y = \delta\,|U| + U_0$ defines a *skew t* distribution with its density function as

$$f(y;\xi,\Sigma,\delta,v) = 2t_{k,v}(y;\xi,\Omega)T_{v+k}\left(\frac{\mu}{\sigma}\sqrt{\frac{v+k}{v+\eta}}\right) \qquad y \in R^k, \tag{4}$$

where

$\Omega = \Sigma + \delta\delta^T$, $\mu = \delta^T\Omega^{-1}(y-\xi)$, $\sigma^2 = (1-\delta^T\Omega^{-1}\delta)$, $\eta = (y-\xi)^T\Omega^{-1}(y-\xi)$.

Here $t_{k,v}(y;\xi,\Omega)$ is the density function of a *k*-dimensional *t* distribution with degrees of freedom *v*(>2), location parameter ξ, and scale matrix Ω, and $T_v(s)$ is the distribution function of a univariate (central) *t* random variable with *v* degrees of freedom.

The multivariate *skew t* mixture model: We consider mixture of distributions whose component distributions belong to the same parametric family, in this case, the *skew t* distribution (4). Using similar notation as above, the mixture probability density function is denoted by

$$f(Y;\Psi) = \sum_{h=1}^{g} p_h f_h(y;\xi_h,\Sigma_h,\delta_h,v_h),$$

where $f_h(y;\xi_h,\Sigma_h,\delta_h,v_h)$ denotes the h^{th} *k*-dimensional *skew t* component with location parameter ξ_h, scale matrix Σ_h, skew parameter δ_h and degrees of freedom v_h. The vector of unknown parameters is denoted by $\boldsymbol{\Psi}$ and can be estimated by maximum likelihood via the EM algorithm[19, 21].

The EM algorithm: For application of the EM algorithm, the observed data vector $(y_1^T,...,y_n^T)^T$ is regarded as incomplete. Using the representation (4) of the *skew t* distribution, we include *u* and *w* as latent unobservable variables. The component-label indicator variables z_{jh} are subsequently introduced, where z_{jh} is defined to be one or zero according to if y_j did or did not arise from the h^{th} component of the mixture model,

(h=1,...,g; j=1,...,n). Letting $z_j = (z_{j1},...,z_{jg})^T$, the complete-data vector x_c is therefore given by $x_c = (x_1^T,...,x_n^T)$ where $x_1 = (y_1^T, u_1^T, w_1^T, z_1^T)^T$ … $x_n = (y_n^T, u_n^T, w_n^T, z_n^T)^T$ are assumed independent and identically distributed with $z_1,...,z_n$ being independent realizations from a multinomial distribution consisting of one draw on g categories with respective probabilities $p_1,...,p_g$. Here $u_j = (u_{j1},...,u_{jg})^T$ and $w_j = (w_{j1},...,w_{jg})^T$. For this specification, the complete-data log likelihood is

$$l_c = \sum_{j=1}^{n}\sum_{h=1}^{g} z_{jh}\{\log(p_h)$$

$$+ \log f_h(y_j \mid w_{jh}, u_{jh})) + \log(\phi(u_{jh}/w_{jh}^{1/2})) + \log(gamma(w_{jh}; v_h/2, v_h/2))\}$$

$$= l_p + \sum_{h=1}^{g} l_h ,$$

where

$$l_p = \sum_{h=1}^{g}\sum_{j=1}^{n} z_{jh}\log(p_h)$$

and

$$l_h = \sum_{j=1}^{n}\{-[\log(|\Sigma_h|) + w_{jh}(y_j - \xi_h - \delta_h \mid u_{jh}|)^T \Sigma_h^{-1}(y_j - \xi_h - \delta_h \mid u_{jh}|)]/2$$

$$-[k\log(2\pi) + k\log(w_{jh})]/2 + [\log(w_{jh}) - \log(2\pi) - w_{jh}u_{jh}^2]/2$$

$$+[-w_{jh}v_h/2 + v_h\log(v_h/2)/2 - \log(\Gamma(v_h/2) + (v_h/2-1)\log(w_{jh})]\}z_{jh}.$$

From the above decomposition, to maximize the Q function of the complete-data log likelihood (McLachlan & Krishnan, 2008), we only need to maximize the functions of l_p and l_h (h=1,2,…,g) separately.

In order to implement the E-step, we calculate the following five conditional expectations, namely,

$\tau_{jh} = P(z_{jh} = 1 \mid y_j)$, $e_{1,jh} = E(w_{jh} \mid y_j, z_{jh} = 1)$, $e_{2,jh} = E(|u_{jh} \mid w_{jh} \mid y_j, z_{jh} = 1)$,
$e_{3,jh} = E(u_{jh}^2 w_{jh} \mid y_j, z_{jh} = 1)$, and $e_{4,jh} = E(\log(w_{jh}) \mid y_j, z_{jh} = 1)$.

These expectations can be calculated using the results that

$$E(z_{jh}w_{jh} \mid y_j) = E(z_{jh}E(w_{jh} \mid y_j, z_{jh}) \mid y_j) = P(z_{jh} = 1 \mid y_j)E(w_{jh} \mid y_j, z_{jh} = 1),$$

$$E(z_{jh} \mid u_{jh} \mid w_{jh} \mid y_j) = P(z_{jh} = 1 \mid y_j)\ E(\mid u_{jh} \mid w_{jh} \mid y_j, z_{jh} = 1),$$

$$E(z_{jh}u_{jh}^2 w_{jh} \mid y_j)) = P(z_{jh} = 1 \mid y_j)\ E(u_{jh}^2 w_{jh} \mid y_j, z_{jh} = 1),$$

$$E(z_{jh}\log(w_j) \mid y_j) = P(z_{jh} = 1 \mid y_j)\ E(\log(w_j) \mid y_j, z_{jh} = 1).$$

The M-step of the EM algorithm maximizes the Q-function of the complete-data log likelihood on each iteration. It follows that the updated estimates of the parameters so obtained on the $(r+1)^{th}$ iteration satisfy

$$p_h^{(r+1)} = \frac{1}{n}\sum_{j=1}^{n} \tau_{jh}^{(r)},$$

$$\xi_h^{(r+1)} = \sum_{j=1}^{n}(y_j e_{1,jh}^{(r)} - \delta_h^{(r)} e_{2,jh}^{(r)})\tau_{jh}^{(r)} / \sum_{j=1}^{n}(\tau_{jh}^{(r)} e_{1,jh}^{(r)}),$$

$$\Sigma_h^{(r+1)} = \{\sum_{j=1}^{n}[(y_j - \xi_h^{(r)})(y_j - \xi_h^{(r)})^T e_{1,jh}^{(r)} - e_{2,jh}^{(r)}\delta_h^{(r)}(y_j - \xi_h^{(r)})^T$$
$$- (y_j - \xi_h^{(r)})\delta_h^{(r)} e_{2,jh}^{(r)T} + e_{3,jh}^{(r)}\delta_h^{(r)}\delta_h^{(r)T}]\tau_{jh}^{(r)}\} / \sum_{j=1}^{n}\tau_{jh}^{(r)},$$

$$\delta_h^{(r+1)} = \frac{\sum_{j=1}^{n}\tau_{jh}^{(r)} e_{2,jh}^{(r)}(y_j - \xi_h^{(r)})}{\sum_{j=1}^{n}(\tau_{jh}^{(r)} e_{3,jh}^{(r)})},$$

$$(\log(v_h^{(r+1)}/2) - \psi(v_h^{(r+1)}/2) + 1)\sum_{j=1}^{n}\tau_{jh}^{(r)} + \sum_{j=1}^{n}(e_{4,jh}^{(r)} - e_{1,jh}^{(r)})\tau_{jh}^{(r)} = 0. \qquad (5)$$

The E and M-steps are alternated repeatedly until the likelihood changes by an arbitrary small amount in the case of convergence.

Singularity problem: As the scale matrices are unconstrained, it is important to consider the problem of relatively large local maxima that occur as a consequence of a fitted component having a very small (but nonzero) generalized variance (the determinant) of the covariance matrices. Such a component converges to a cluster containing a few data points either relatively close together or lying in almost a lower dimensional subspace in the case of multivariate data.

Methods to estimate the degrees of freedom: The solution of equation (5) for the updated estimate of the degrees of freedom for the h^{th} component does not exist in closed form. We provide three options for its computation. With the first option, we use an approximation to the term $e_{4,jh}^{(r)}$ on the right-hand of equation (5). With the second option, the term $e_{4,jh}^{(r)}$ is calculated by truncating an infinite series expansion of it. Finally, for the third option, we do not estimate the degrees of freedom for the components, but instead we specify their values beforehand. A comparison of the three options in some simulation experiments suggest that Option 1 (which is quicker) performs not too far short of Option 2, which attempts to provide the exact values at each stage of the iterative process.

Model selection criteria

To determine the optimal number of components (g*) in the mixture model, FLAME uses by default a novel Scale-free Weighted Ratio (SWR) criterion (for details see Methods, main text). The more commonly used unweighted ratio of average intra- to intercluster distances does not distinguish between the distinct scale variances of different components or between outlier and non-outlier cells. SWR addresses this with the following strategies: first, Mahalanobis distance is used since it normalizes Euclidean distance by the scale variance of the distribution of points, thus it is independent of dispersion levels that vary from one population to another. Also, Mahalanobis distance has the desirable property of being invariant to all non-singular transformations. Second, by using the posterior probabilities as weights, it restricts the influence of outlier cells on the determination of the optimal number of populations. The combined effect of these two strategies allows SWR to perform robust and accurate model selection. An unweighted version of SWR, which is average Intracluster Euclidean distance to average Intercluster Euclidean distance Ratio (IIR), is another option that is available in FLAME:

$$IIR = \frac{\sqrt{\sum_{i,j \in C} d^2(i,j)} \Big/ \left|\left\{(i,j): i,j \in C\right\}\right|}{\sqrt{\sum_{i \in C, j \in C', C \neq C'} d^2(i,j)} \Big/ \left|\left\{(i,j): i \in C, j \in C', C \neq C\right\}\right|} \qquad (6)$$

where $d(i, j)$ is the Euclidean distance between two points i and j. Note that IIR is a special case of SWR, which assumes all scale variances and posterior probabilities to be equal. It allows faster computation and is suitable for very large samples with well-separated clusters.

FLAME also computes the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC) and Integrated Completed Likelihood (ICL), which are available as alternate criteria for model selection. For these information criteria, we choose optimal number of components g* by considering the likelihood function. In the absence of any prior information as to the number of clusters present in the data, we can monitor the increase in log likelihood function as the value of g increases. At any stage, the choice of $g = g_0$ versus $g = g_0 + 1$ can be made by either performing the likelihood ratio test or by using some information-based criteria such as AIC or BIC. Unfortunately, regularity conditions do not hold for the likelihood ratio test statistic λ to have its usual null distribution of χ^2 with degrees of freedom equal to the difference d in the number of parameters for $g = g_0 + 1$ and $g = g_0$ components in the mixture model. With information criteria such as AIC and BIC, we choose $g = g_0 + 1$ over $g = g_0$ if $-2\log(\lambda)$ is greater than cd, where, corresponding to the use of AIC and BIC, c is equal to 2 and $\log(n)$, respectively. The ICL attempts to improve the performance of AIC and BIC by replacing $d \log(n)$ in the use of BIC by $d \log(n) + 2\hat{e}$, where $\hat{e}$ is the plug-in estimate of the entropy, which is given by

$$\hat{e} = -\sum_{j=1}^{n} \sum_{h=1}^{g} \tau_{jh} \log(\tau_{jh}) \qquad (7)$$

where τ_{jh} is the posterior probability that the j^{th} data point belongs to the h^{th} component of the mixture (h=1,…,g; j=1,…,n); see McLachlan and Peel (2000; Chapter 6). To

choose between one or more than one components, FLAME uses BIC as the default criterion since intercluster distances, as in SWR, are not defined for one component. In summary, the different model selection criteria that are offered by FLAME are as follows: SWR (by default), IIR, BIC, AIC and ICL.

Metaclustering

Details of Metaclustering Method

As outlined in the Methods section of the paper, we developed a novel 2-step strategy to match corresponding populations across samples.

Step 1: *PAM clustering* – In this step, we do robust clustering of the samples within a particular class to generate a high-dimensional template marking the typical locations and weights (i.e. proportion of cells) of the populations of that class. The clustering is done with PAM[22], which is a robust version of the k-means algorithm, and the resulting typical populations for that class are called *metaclusters*. A metacluster's location is given by the corresponding PAM medoid. A metacluster's weight is given by median size of the *c* clusters that are nearest to its location and belong to it *c* is, by default, set to 20% of the number of clusters in a metacluster). The optimal number of metaclusters in the template is determined by average silhouette width[23].

Step 2: *Bipartite Matching* – In this step, we take a graph optimization approach to solve the metaclustering problem by optimally matching every sample to its class-template computed in step 1. For this purpose, we model the problem as an enhanced version of the minimum cost bipartite matching problem from graph theory[24], which we describe below.

A *bipartite* graph $G = (S, T; E)$ consists of two disjoint sets of *nodes* S and T and a set E of node-pairs, or *edges*, such that for every edge (s, t) in E one node (s) is from S and the other (t) from T in which case we say that the nodes s and t are "matched". Also, every edge (s, t) can have a cost of matching d_{st} associated with it. Further, we also define the *capacity* of a node, denoted by p_s or q_t for $s \in S$ and $t \in T$, and require that for a node (say $s \in S$) that is matched to one or more nodes (say $T' \subseteq T$), their capacities should also "match", i.e. $p_s = \sum_{t \in T'} q_t$. To allow approximate matching of capacities, the latter condition may be relaxed as $\sum_{t \in T'} q_t - p_s \leq \varepsilon$ for a pre-specified small value ε.

We formulate metaclustering as finding a minimum cost bipartite matching problem following additional capacity matching constraints. As stated above, within-class metaclustering involves matching a sample's modes (S) with the template (T) of that class specified by metacluster locations (the PAM medoids), while cross-class metaclustering involves matching the templates of two classes. We use the within-class Euclidean distance between a mode s and a medoid t as the cost function d_{st} and the corresponding weights p_s and q_t as the node capacities for S and T.
To solve the above constrained bipartite matching problem, we use the following integer programming (IP) formulation: given the distances d_{ij} for all mode-medoid pairs and the capacities p_i and q_j for all modes and medoids respectively, we want to compute the optimal matching, represented by binary variables $\{x_{ij}\}_{i \in S, j \in T}$ such that $x_{ij} = 1$ if and only if mode i is matched to medoid j, as a solution to the IP:

Minimize $\sum_{i \in S, j \in T} d_{ij} x_{ij}$ (8)

subject to the following constraints

$\sum_{i \in S} p_i x_{ij} \le q_j + \varepsilon$ for all $j \in T$ (9a)

$\sum_{j \in T} q_j x_{ij} \le p_i + \varepsilon$ for all $i \in S$ (9b)

$\sum_{i \in S} x_{ij} \ge 1$ for all $j \in T$ (10a)

$\sum_{j \in T} x_{ij} \ge 1$ for all $i \in S$ (10b)

$x_{ij} \in \{0,1\}$ for all $i \in S$, $j \in T$ (11)

where ε is a pre-specified constant (by default set to 0.05) that allows approximate matching of node capacities.

By default, we run the IP solver *lpSolve*[25] with all the constraints (9a-10b) mentioned in the formulation. However certain constraint-relaxations can allow optimal matchings when no feasible solution might exist for the default formulation. For instance, the removal of the constraint (10a) allows one or more template medoids to be left unmatched (for instance, to allow detection of a missing population in a sample); the removal of constraints (9a)-(9b) allows the node capacities to be ignored and then the matching is purely based on distances between the modes and the medoids.

For cross-class metaclustering, we first apply bipartite matching to the templates of the two compared classes and then extend the matching from the templates to the samples based on the results of within-class matching as described above.

When called from our R program, the IP solver lpSolve performed all computations for the present application efficiently, e.g. for metaclustering 20 samples in a typical run in the simulation study in Part IV.b, the average running time was 0.393 sec with a standard deviation of 0.023 sec). The IP solver is capable of handling up to 100 modes or medoids, which makes our metaclustering algorithm scalable for samples with large number of populations.

Assessment of Metaclustering Stability

Simulation experiment design: To demonstrate the stability of the new 2-step metaclustering method, we performed 100 metaclustering runs, or trials, each with an input of a subset of samples drawn randomly, without replacement, from a pool of real

flow cytometric data. We observed that our metaclustering algorithm yielded consistent results across 99/100 trials.

The dataset consisted of the 30 samples from our T cell phosphorylation dataset (see Supplementary Methods, above) that were collected 5-minute after anti-CD3 stimulation. Each sample was collected from a different individual, but they all belong to the same phenotypic (post stimulation) class. Prior to metaclustering, each sample was modeled with multivariate *skew t* distributions over a range of 4-6 clusters, and the optimal number of clusters was selected using the SWR criterion. During each trial, 20 of the 30 samples were selected at random, without replacement, to participate in the metaclustering.

During each trial, metaclustering was performed as described in the Methods section (see main text). The range of metacluster count was taken to be 5-6, since the optimal *g* of each sample at the end of clustering was either 5 or 6. The optimal number of metaclusters was determined by maximizing the average silhouette width[23] (ASW).

Results: In 99 trials, k=5 was the preferred optimal number of metaclusters. In trial #39, k=6 is preferred to k=5 based on ASW values that differed by 0.00052, so the two models were essentially equivalent in this trial. The associated average silhouette widths are shown in Figure 1 below. Paired *t*-tests showed that k=5 is preferred to k=6 by a difference in ASW median value of 0.0297 (*P* value = 4.11×10^{-39}) rendering k*=5 as the optimal number of clusters for this dataset. The full table of ASW values for each trial is shown in Table 2 below.

Because all 30 samples participating in the trials belong to the same phenotypic class, we expect the configurations of metaclusters to be consistent across trials. Figures 2(a)-(f) below are 3D projections (SLP76, CD4, CD45RA dimensions) of metacluster assignments of six out of the 100 trials performed, and they clearly illustrate the consistency of the metaclustering results.

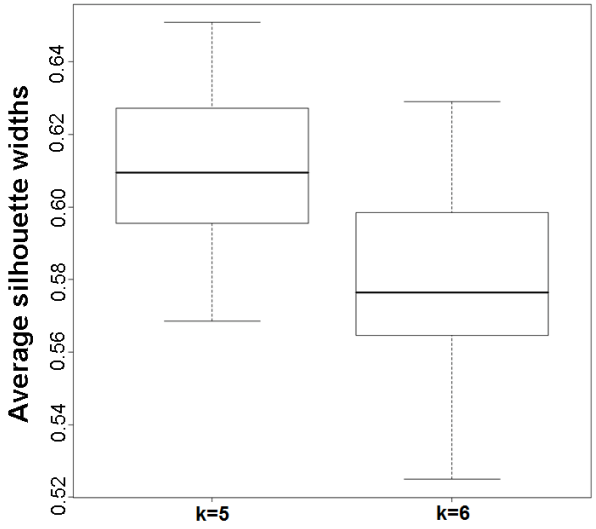


Figure 1. Average silhouette widths for the k=5 and k=6 solutions to the clustering of random subsets of 20 samples from a pool of 30 samples. The median ASW values are estimated from 100 trials (black line). The box defines the interquartile range, and the ends of the whiskers define the lowest and highest non-outlier values (no outlier observed). k=5 is the optimal solution in 99 of the 100 individual trials.

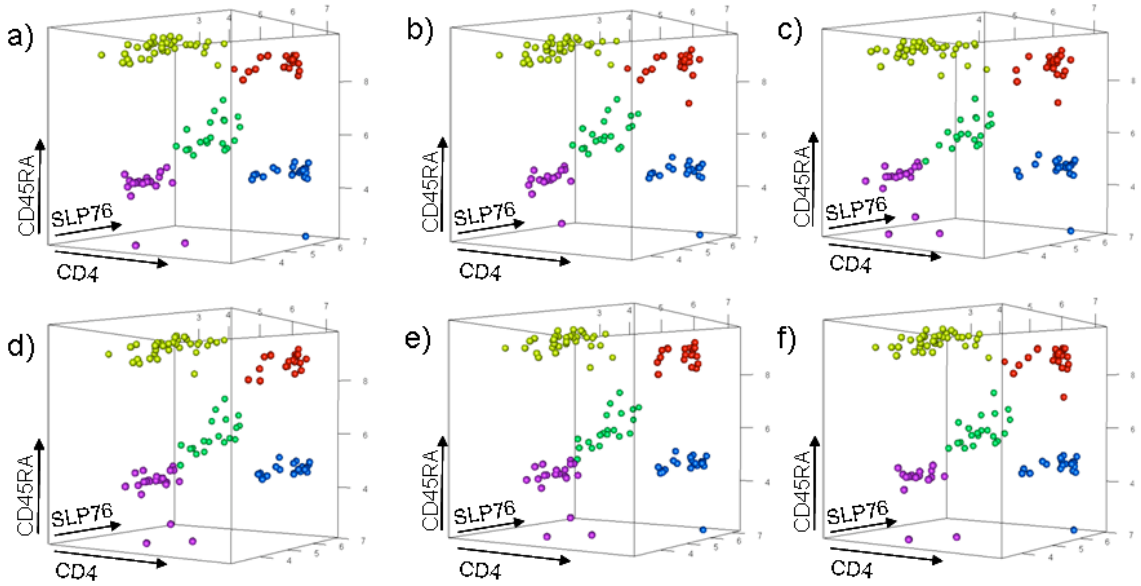


Figure 2. 3D projections of metacluster configuration in six representative trials. The presented metaclusters were drawn arbitrarily from the 100 trials:(a) Trial #10; (b) Trial #20; (c) Trial #30; (d) Trial #40; (e) Trial #50; (f) Trial #60. Each plot shows the metaclustering results on the pooled collection of the clusters participating in a particular trial. Each point represents the mode of one cluster from one individual sample. Every metacluster has its own distinct color and the cluster modes (represented by dots) which belong to it have that color.

Table 1. Average Silhouette Width score for k=5 and k=6 in 100 metaclustering trials.

Trial #	k=5	k=6	Trial #	k=5	k=6	Trial #	k=5	k=6	Trial #	k=5	k=6
1	0.609	0.594	**26**	0.580	0.535	**51**	0.634	0.618	**76**	0.587	0.566
2	0.631	0.615	**27**	0.631	0.606	**52**	0.595	0.576	**77**	0.599	0.572
3	0.577	0.525	**28**	0.631	0.613	**53**	0.612	0.580	**78**	0.616	0.557
4	0.621	0.604	**29**	0.627	0.600	**54**	0.619	0.599	**79**	0.627	0.587
5	0.599	0.560	**30**	0.646	0.625	**55**	0.629	0.587	**80**	0.638	0.594
6	0.610	0.598	**31**	0.600	0.584	**56**	0.633	0.603	**81**	0.631	0.609
7	0.569	0.546	**32**	0.599	0.548	**57**	0.632	0.610	**82**	0.588	0.559
8	0.589	0.568	**33**	0.627	0.613	**58**	0.605	0.572	**83**	0.576	0.575
9	0.631	0.629	**34**	0.628	0.587	**59**	0.608	0.569	**84**	0.595	0.561

10	0.633	0.621	**35**	0.582	0.562	**60**	0.643	0.609	**85**	0.608	0.597
11	0.609	0.568	**36**	0.611	0.573	**61**	0.616	0.581	**86**	0.600	0.572
12	0.651	0.620	**37**	0.612	0.566	**62**	0.616	0.559	**87**	0.587	0.540
13	0.594	0.564	**38**	0.575	0.556	**63**	0.575	0.537	**88**	0.615	0.570
14	0.641	0.587	**39**	0.608	0.609	**64**	0.602	0.545	**89**	0.620	0.577
15	0.577	0.559	**40**	0.634	0.593	**65**	0.589	0.545	**90**	0.625	0.598
16	0.595	0.568	**41**	0.645	0.601	**66**	0.603	0.596	**91**	0.600	0.572
17	0.614	0.584	**42**	0.578	0.567	**67**	0.590	0.572	**92**	0.601	0.576
18	0.586	0.545	**43**	0.626	0.605	**68**	0.580	0.536	**93**	0.599	0.547
19	0.602	0.569	**44**	0.633	0.622	**69**	0.596	0.568	**94**	0.616	0.598
20	0.619	0.577	**45**	0.588	0.530	**70**	0.612	0.565	**95**	0.624	0.599
21	0.598	0.570	**46**	0.623	0.586	**71**	0.578	0.559	**96**	0.616	0.563
22	0.573	0.549	**47**	0.605	0.582	**72**	0.630	0.602	**97**	0.610	0.587
23	0.647	0.614	**48**	0.609	0.570	**73**	0.610	0.580	**98**	0.597	0.577
24	0.597	0.574	**49**	0.589	0.567	**74**	0.629	0.611	**99**	0.629	0.626
25	0.630	0.584	**50**	0.618	0.576	**75**	0.609	0.591	**100**	0.596	0.557

Supplementary Discussion

Performance Analysis

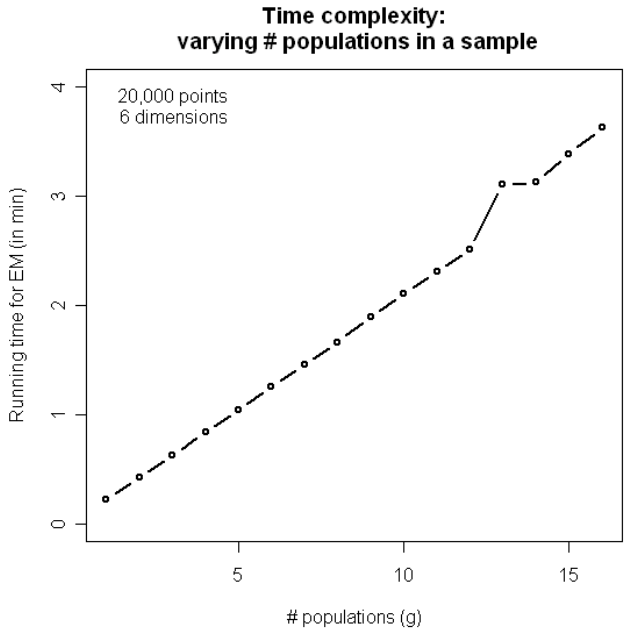
Simulation studies were done to evaluate the time complexity of the EM algorithm for multivariate *skew t* mixture modeling that forms the core of FLAME. Factors affecting the total computing time of the EM algorithm include the number of observations n (a.k.a. sample size), the number of dimensions p, the number of clusters/populations g, and the total number of iterations of the EM routine. It is important to note that while the first three factors are pre-determined, the last one is variable for each dataset. Upon simulation, the computing times for the E step and the M step were recorded for each setting and then added together. The computing times are based on a desktop running Windows XP Pro (2002) with the following specifications: Intel(R) Pentium(R) 4 processor with 2.4GHz, 504 MB of RAM.

First, we observed that the time complexity is linear in the number of clusters/populations (g) while keeping the values of n and p fixed at 20,000 and 6 respectively (Figure 3a and Table 2, below). In the next simulation study, the number of EM iterations was fixed at 50 and g at 3, and n was varied from 10,000 to 100,000 while incrementing in steps of 1,000, and p was varied from 3 to 20. In figure 3(b), we note that for fixed number of n=50,000 and 100,000 observations (purple and green points respectively), the running time increases, in both cases, quadratically with respect to p (smoothed curve shown in dashed line). In figure 3(c), we note that for fixed dimension p=10 and 20 (blue and red points respectively), the running time increases, in both cases, linearly with respect to n. Notably, both trends are projections from a common set of running times (Table 3, below) specifying a joint trend with respect to tuple (n,p). To summarize, as is well known, the total computing time appears to be linear in n and g, and quadratic in p.

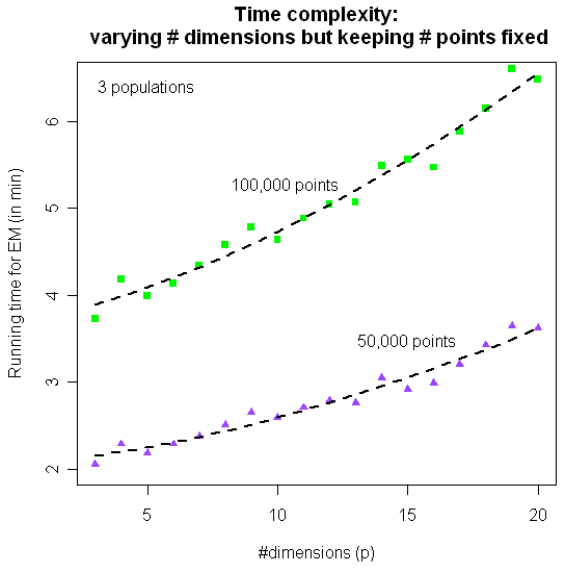
Given the linear (for increase in numbers of populations or cells/points) and gentle quadratic (for increase in numbers of dimensions) trends in our performance analysis, FLAME scales well as we challenge it with larger and more complex datasets. In addition, it is worth noting that it compares favorably with a human operator: for example, it is able to process 50,000 cells/points belonging to 3 populations in 10 dimensions of information in ~2.5 minutes (See Figure 3, below).

Table 2. Time complexity for increasing ranges of populations to model. By varying the number of cell clusters (*g*, first row) for *skew t* mixture modeling of 6-dimensional simulated data with 20,000 points, the running time for EM (in minutes) for a given value of *g*, and the cumulative running time (in min) for the range 1 through *g*, are tabulated in the second and third rows respectively.

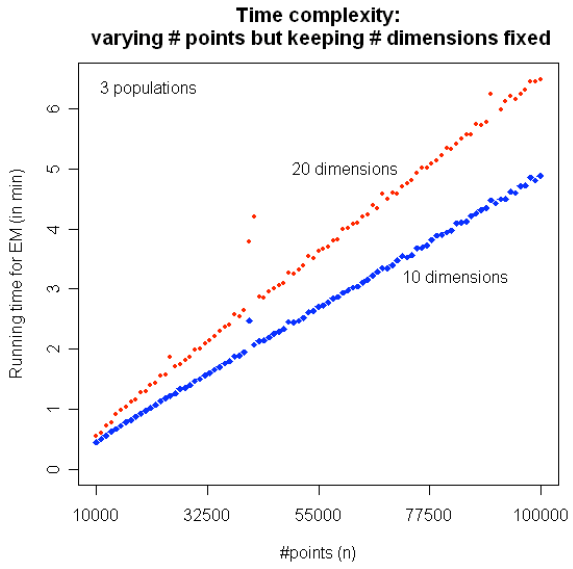
g	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Runtime to model g clusters (min)	0.22	0.42	0.63	0.84	1.04	1.25	1.46	1.66	1.9	2.1	2.31	2.52	3.11	3.13	3.39	3.63
Cumulative Runtime to determine optimal g (min)	0.22	0.64	1.27	2.11	3.14	4.4	5.85	7.51	9.41	11.5	13.8	16.3	19.4	22.6	26	29.6



(a)



(b)



(c)

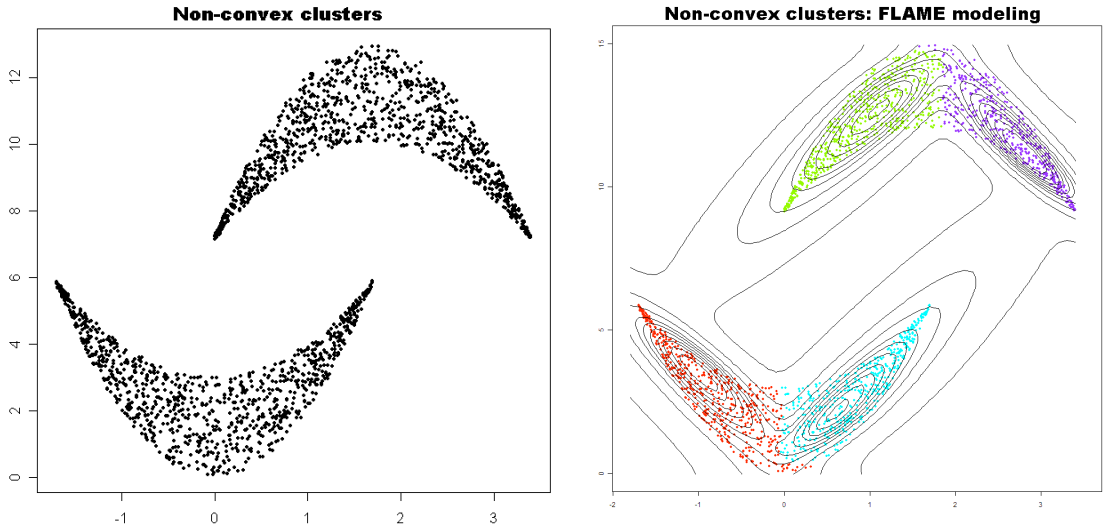
Figure 3. Time complexity trends. Running times in minutes as a result of varying (a) number of populations, (b) number of dimensions, (c) number of points in an input sample to multivariate *skew t* mixture modeling. The trends appear to be linear in (a) and (c) and quadratic in (b).

Table 3. Simulation results on EM time complexity. Each entry in the table shows the running times (in minutes) for multivariate *skew t* mixture modeling of simulated samples with different number of points (rows) and dimensions (columns); the number of populations was fixed at 3. Simulations were run with increments of 1000 points, and results ar plotted in Figure 3. For conciseness, we present only increments of 5000 in the table below.

	# dimensions																	
	3	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
#points																		
10k	0.37	0.41	0.39	0.4	0.42	0.44	0.45	0.43	0.45	0.46	0.45	0.49	0.46	0.46	0.49	0.53	0.55	0.55
15k	0.56	0.63	0.6	0.61	0.65	0.68	0.77	0.69	0.72	0.75	0.75	0.82	0.78	0.81	0.87	0.91	0.98	0.98
20k	0.74	0.83	0.8	0.82	0.87	0.91	0.96	0.93	0.97	1.02	1.02	1.11	1.06	1.1	1.19	1.23	1.33	1.29
25k	0.93	1.03	0.99	1.04	1.07	1.14	1.21	1.16	1.22	1.26	1.27	1.39	1.29	1.38	1.47	1.55	1.67	1.87
30k	1.12	1.25	1.19	1.24	1.31	1.38	1.48	1.41	1.47	1.54	1.51	1.69	2.3	1.64	1.81	1.86	1.97	1.98
35k	1.3	1.46	1.4	1.44	1.51	1.6	1.67	1.61	1.7	1.77	1.75	1.94	1.85	1.92	2.07	2.14	2.33	2.29
40k	1.49	1.67	1.59	1.66	1.74	1.83	1.93	1.86	1.95	2.04	2.03	2.21	2.12	2.19	2.36	2.48	2.64	2.64
45k	1.67	1.88	1.8	1.87	1.95	2.05	2.17	2.11	2.19	2.27	2.28	2.49	2.37	2.46	2.66	2.77	2.96	2.96
50k	1.86	2.07	1.99	2.09	2.17	2.29	2.41	2.35	2.44	2.54	2.53	2.77	2.62	2.71	2.93	3.14	3.36	3.26
55k	2.05	2.29	2.19	2.29	2.38	2.51	2.65	2.59	2.71	2.79	2.76	3.05	2.91	2.99	3.21	3.43	3.65	3.62
60k	2.23	2.49	2.37	2.48	2.59	2.74	2.87	2.79	2.94	3.08	3.06	3.34	3.22	3.31	3.59	3.8	4.03	3.99
65k	2.42	2.7	2.59	2.69	2.8	2.97	3.11	3.01	3.16	3.29	3.28	3.57	3.44	3.52	3.79	4	4.31	4.24
70k	2.6	2.91	2.79	2.9	3.02	3.2	3.39	3.25	3.4	3.56	3.58	3.9	3.72	3.86	4.14	4.32	4.69	4.6
75k	2.79	3.12	3	3.1	3.24	3.42	3.61	3.47	3.67	3.8	3.8	4.13	3.95	4.11	4.38	4.61	5	4.94
80k	2.98	3.33	3.18	3.3	3.46	3.65	3.88	3.7	3.9	4.05	4.05	4.42	4.24	4.4	4.72	4.92	5.39	5.23
85k	3.18	3.54	3.41	3.51	3.68	3.88	4.52	3.94	4.13	5.08	4.3	6.16	4.5	4.63	5.03	5.27	5.66	5.58
90k	3.35	3.77	3.59	3.74	3.89	4.15	4.42	4.23	4.48	4.64	4.63	5.18	4.9	5.14	5.5	5.65	6.21	6.25
95k	3.54	4.01	3.81	3.91	4.1	4.33	4.59	4.39	4.6	4.79	4.82	5.28	5.05	5.21	5.52	5.9	6.34	6.15
100k	3.73	4.18	3.99	4.13	4.35	4.58	4.78	4.64	4.89	5.04	5.07	5.49	5.56	5.47	5.88	6.15	6.61	6.48

Nonconvex clustering

Although in general a finite mixture modeling algorithm identifies convex components, it could be used for modeling a nonconvex cluster by combining more than one sufficiently overlapping convex components. Using the mixture model parameters such as component locations and variances estimated by EM, an efficient methodology to model nonconvex clusters is described in Mitra *et al*[26]. The same technique could be applied downstream of FLAME *skew t* mixture modeling. In fact, FLAME's use of Mahalanobis distance is particularly suitable for determination of overlapping components. Furthermore, formal modeling of skew by FLAME helps to accurately identify asymmetric components thus guarding against spurious overlap (as noted in the paper, spurious overlap is possible for symmetric Gaussian or *t* mixture due to skew and not any nonconvexity of data). When a nonconvex cluster is split into convex components, the latter are often asymmetric; therefore *skew t* distributions can naturally lead to an optimal model. Figure 4 below shows FLAME modeling of a simulated nonconvex sample with two pairs of overlapping *skew t* components. However, while all ingredients for modeling nonconvex clusters are computed by FLAME, in the present version of the package we did not include the modality to directly output the nonconvex clusters. An expert user can model a nonconvex cluster herself using a sub-mixture of multiple overlapping components based on the parameters estimated by FLAME where the overlap can be determined with Mahalanobis distance between component locations[26].

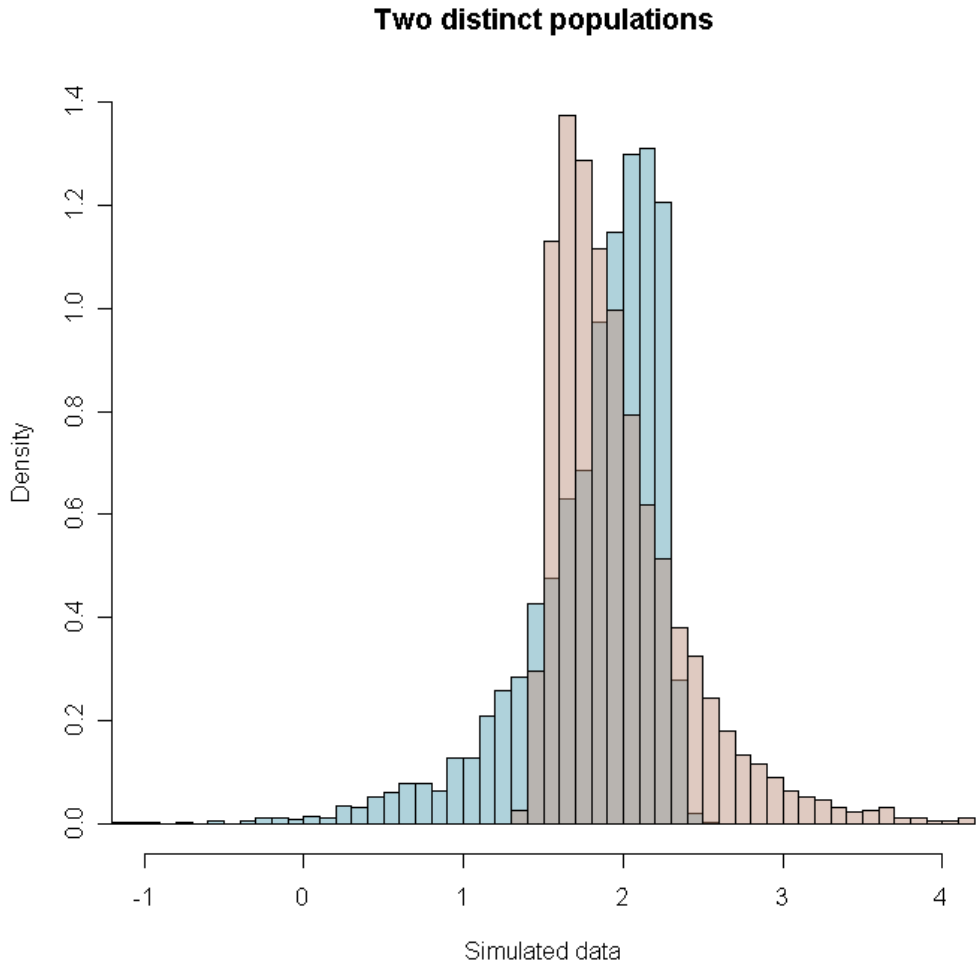


(a) (b)

Figure 4. Nonconvex clustering with FLAME. A sample with 2 nonconvex clusters in plot (a) is modeled by FLAME with 4 *skew t* components as shown in plot (b). Two sub-mixtures of *skew t* distributions, represented by the overlapping green-purple and red-turquoise component-pairs in plot (b), are used for modeling of the two nonconvex clusters in plot (a).

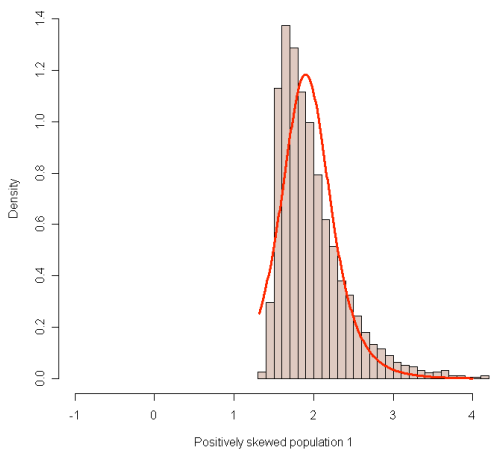
Comparison of Box-Cox Transformation and Symmetric *t* with Direct Use of *Skew t* Distribution for Modeling

Lo et al., in their method implemented in flowClust[27], transform flow cytometric data to minimize skew and then model the transformed data with symmetric *t* distributions. In contrast our method directly models the asymmetric populations with *skew t* distributions and thus learns the distinctive shape and location of each population. In Figure 5(a) below, two hypothetical 1-dimensional populations with distinct shapes and modes are shown. The blue histogram is a right-skewed population and the brown histogram is a left-skewed population. Figures 5(b) and 5(c) show the Box-Cox transformed, symmetric *t* density fit of the left-skewed and the right-skewed distributions with the solid blue and red lines respectively. Figures 5(d) and 5(e) show the *skew t* density fit by FLAME of the left-skewed and the right-skewed distributions with the solid blue and red lines respectively. When the modeling results of Lo et al. and our method for the individual populations are superimposed, as seen in figures 5(f) and 5(g), we note that that the Box-Cox symmetric *t* modeling of these two very distinct populations are almost indistinguishable (Figure 5(f)), whereas the two *skew t* modeled populations are clearly distinguishable in Figure 5(g).

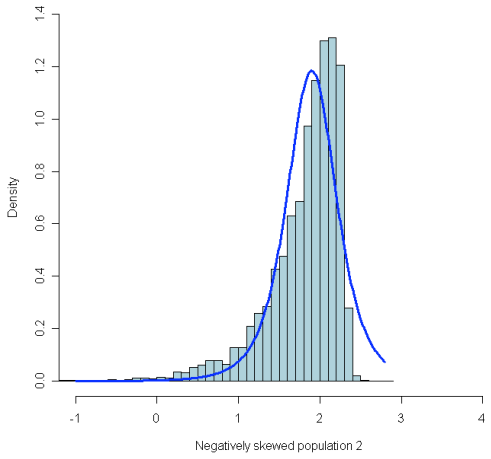


(a)

Box-Cox and *t* fit by flowClust

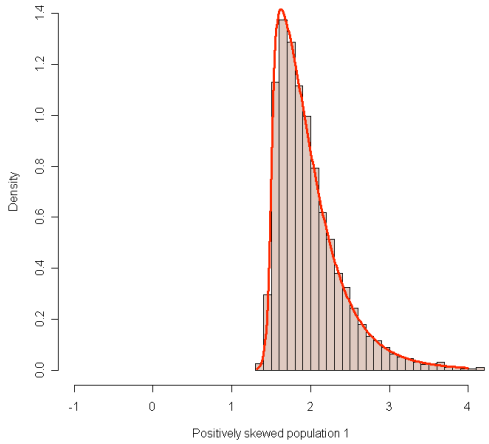


(b)

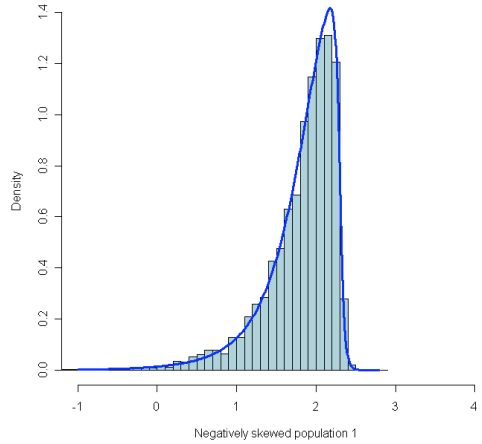


(c)

Skew *t* fit by FLAME

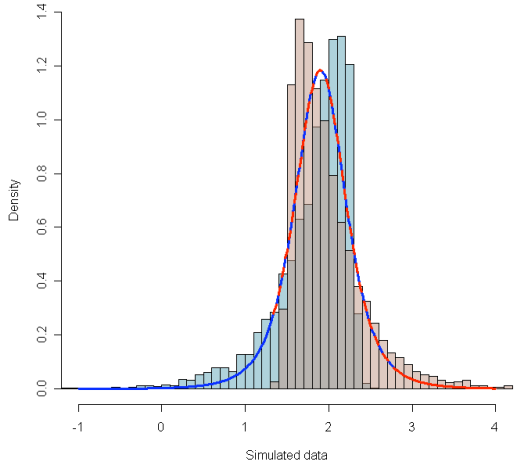


(d)



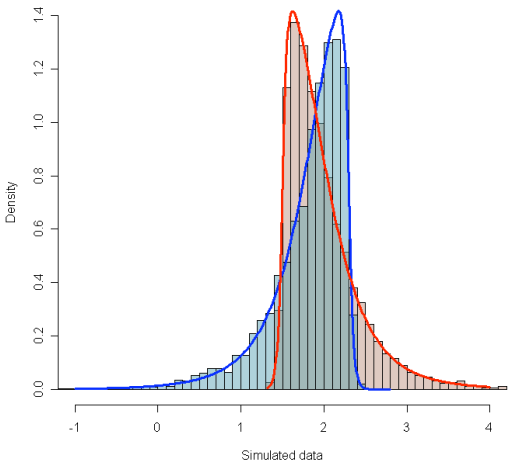
(e)

Overlapping fit by flowClust



(f)

Distinct fit by FLAME



(g)

Figure 5. Comparative modeling of distinct skew populations by our method and by Lo et al. (a) presents our simulated data which consists of skew distributions that are mirror images of each other. (b) and (c) present the individual results of modeling each distribution using flowClust[27]. (d) and (e) are the solutions computed by FLAME. Following FlowClust modeling, when we overlay the two FlowClust solutions, we see that they are fitted with the same model (f): this approach therefore fails to appropriately distinguish the skew found in each set of data. On the other hand, in (g), we see that FLAME is able to accurately model the unique properties of each set of data.

References

1. Parks, D.R., Roederer, M. & Moore, W.A. A new "Logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. *Cytometry A* **69**, 541-551 (2006).
2. Baecher-Allan, C., Wolf, E. & Hafler, D.A. MHC class II expression identifies functionally distinct human regulatory T cells. *J Immunol* **176**, 4622-4631 (2006).
3. www.flowjo.com
4. Maier, L.M., Anderson, D.E., De Jager, P.L., Wicker, L.S. & Hafler, D.A. Allelic variant in CTLA4 alters T cell phosphorylation patterns. *Proceedings of the National Academy of Sciences of the United States of America* **104**, 18607-18612 (2007).
5. McLachlan, G.J. & Basford, K.E. Mixture models. Inference and applications to clustering. (Dekker, New York; 1988).
6. McLachlan, G.J. & Peel, D. Finite Mixture Models. (Wiley, New York, 2000).
7. Frühwirth-Schnatter, S. Finite Mixture and Markov Switching Models. (Springer, New York; 2006).
8. Lange, K.L., Little, R.J.A. & Taylor, J.M.G. Robust Statistical Modeling Using the *t* Distribution. *Journal of the American Statistical Association* **84**, 881 (1989).
9. Azzalini, A. A class of distributions which includes the normal ones. *Scand. J. Statist* **12**, 171-178 (1985).
10. Azzalini, A. & Capitanio, A. Statistical applications of the multivariate skew normal distribution. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **61** (1999).
11. Azzalini, A. & DallaValle, A. The multivariate skew-normal distribution. *Biometrika* **83**, 715-726 (1996).
12. Gupta, A.K. Multivariate skew *t*-distribution. *Statistics* **37**, 1 (2003).
13. Gupta, A. & Chen, J. A class of multivariate skew-normal models. *Annals of the Institute of Statistical Mathematics* **56**, 305 (2004).
14. Azzalini, A. & Capitanio, A. Distributions generated by perturbation of symmetry with emphasis on a multivariate skew *t*-distribution. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65** (2003).
15. Sahu, S.K., Dey, D.K. & Branco, M.D. A new class of multivariate skew distributions with applications to Bayesian regression models. *Canadian Journal of Statistics-Revue Canadienne De Statistique* **31**, 129-150 (2003).

16. Banfield, J.D. & Raftery, A.E. Model-Based Gaussian and Non-Gaussian Clustering. **49**, - 821 (1993).
17. Lin, T., Lee, J. & Hsieh, W. Robust mixture modeling using the skew *t* distribution. *Statistics and Computing* **17**, 81-92 (2007).
18. Lin, T.I., Lee, J.C. & Yen, S.Y. Finite Mixture Modeling using The Skew Normal Distribution. *Statistica Sinica* **17**, 909-927 (2007).
19. Dempster, A.P., Laird, N.M. & Rubin, D.B. Maximum Likelihood from Incomplete Data via the *EM* Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39**, 1 (1977).
20. Arellano-Valle, R.B., Genton, M.G. & Loschi, R.H. Shape mixtures of multivariate skew-normal distributions. *Journal of Multivariate Analysis* **In Press, Corrected Proof** (2008).
21. McLachlan, G.J. & Krishnan, T. The EM algorithm and extensions, Edn. 2nd. (Wiley-Interscience, Hoboken, N.J.; 2008).
22. Kaufman, L. & Rousseeuw, P.J. Finding groups in data: an introduction to cluster analysis. (Wiley, New York; 1990).
23. Rousseeuw, P.J. Silhouettes - a Graphical Aid to the Interpretation and Validation of Cluster-Analysis. *J Comput Appl Math* **20**, 53-65 (1987).
24. Korte, B.H. & Vygen, J. Combinatorial optimization: theory and algorithms, Edn. 3rd. (Springer, Berlin; New York; 2006).
25. Buttrey, S.E. Calling the lp_solve linear program software from R,S-PLUS and Excel. *J Stat Softw* **14**, - (2005).
26. Mitra, P., Pal, S.K. & Siddiqi, M.A. Non-convex clustering using expectation maximization algorithm with rough set initialization. *Pattern Recogn Lett* **24**, 863-873 (2003).
27. Lo, K., Brinkman, R.R. & Gottardo, R. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry A* **73**, 321-332 (2008).